

Defining Mathematical functions

Defining function - Anonymous functions

- A function that is *not* stored in a program file, but is associated with a variable whose data type is `function_handle`, e.g
- Scalar anonymous function

```
1      >> f1=@(x) cos(x).*sin(2*x)
2      f1 =
3          @(x)cos(x).*sin(2*x)
4      >> f1(-1)
5      ans =
6          -0.4913
```

Anonymous functions

- 2D anonymous function

```
1     >> g1=@(x,y) sin(x.^2 + y.^2)./(x.^2 +y.^2)
2     g1 =
3         @(x,y) sin(x.^2+y.^2)./(x.^2+y.^2)
4     >> g1(0.01,0.01)
5     ans =
6         1.0000
```

fplot(f,xinterval)

- `fplot(f,xinterval)` - plots $y=f(x)$ over the specified interval

```
1         f = @(x) sin(8*x);  
2         fplot(f,[0 2*pi],'r');  
3         title('$sin(8x)$','Interpreter','latex')
```

- `fplot` has a relative error tolerance of $2e-03$

Indirect function evaluation

$$[y_1, \dots, y_N] = \text{feval}(\text{fun}, x_1, \dots, x_M)$$

- Evaluates a function using its name or its handle using inputs x_1, \dots, x_M , e.g

```
1      >> feval('cos',0.5)
2      ans =
3      0.8776
```

```
1      >> g=inline('sin(x.^2 + y.^2)./(x.^2 +y.^2)');
2      >> feval(g,0.01,0.01)
3      ans =
4      1.0000
```

Indirect function evaluation

```
[y1, . . . , yN] = feval(fun, x1, . . . , xM)
```

- Evaluates a function using its name or its handle using inputs x_1, \dots, x_M , e.g

```
1 >> g1=@(x,y) sin(x.^2 + y.^2)./(x.^2 +y.^2);  
2 >> feval(g1,0.01,0.01)  
3 ans =  
4      1.0000
```

- Note: if you define your own scalar function in a file `f1.m` you can also use `feval` to evaluate it as `>>feval('f1',x)`

Other useful commands

Command	Result
<code>fplot('f',[x1, x2])</code>	plots the <code>function</code> <code>f</code> on <code>[x1,x2]</code>
<code>fminbnd('f',x1,x2)</code>	returns <code>x</code> , a local min of <code>f</code> on <code>[x1,x2]</code>
<code>fzero('f',x0)</code>	returns the root of <code>f</code> near <code>x0</code>